UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/575,342	05/19/2000	Jouni Rapakko	460-009420-US(PAR)	9989

7590   01/03/2007

Clarence A Green
Perman & Green LLP
425 Post Road
Fairfield, CT 06430

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2193	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

	Application No.	Applicant(s)
Office Action Summary	09/575,342	RAPAKKO ET AL.
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>11 October 2006</u>.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-31</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-31</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☒ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. <u>12/15/06</u>.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

1.   This action is responsive to the Applicants'response filed 10/11/2006.

As indicated in Applicants'response, claims 1, 7, 14-15, 21, and 28 have been amended

and claims 29-31 added. Claims 1-31 are pending in the office action.

Claim Rejections - 35 USC § 112

2.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the
> subject matter which the applicant regards as his invention.

3.   Claims 7-31 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for

failing to particularly point out and distinctly claim the subject matter which applicant regards as

the invention.

Claims 7, 14-15, 21, 28 recite 'being separate parts of the user interface software ...' in

conjunction with 'stored in a memory of the electronic device' (cl. 7); '... stored within the

storage' (cl. 14); and '...stored in the electronic device'(cl. 15, 21, 28); the confusing

relationships can be described as following:

(i) when it is construed that the user interface software has two parts, the basic module

and the user interface module, the reciting that both parts *are stored* within the electronic device'

does not make it clear what actually has been currently stored in the electronic device system

when the step of loading and starting occurs (e.g. *executing the loading...*) in light of the role by

the basic module to load the UI module, both the basic and the UI module associated to the

expansion card UI software. That is, this 'being stored' or 'are stored' leads to more than one

interpretations; e.g. EITHER both parts of the user interface (UI) software are already resident

but unrelated to this loading/starting; OR that they are currently resident in the system for the

loading/starting to actualize; OR that the loading would have for result that these parts (UI module and basic module) be stored in the device. For example, there is vagueness in relationship between 'executing the loading' (e.g. claim 15) and 'wherein the user interface software comprises ... separate parts ... stored within the electronic device'. The loading is organized in 2 phases involving 2 separate UI parts, one part helping the other part to be loaded. The sequence needed to fulfill the *loading* of 2 separate parts (which entails separation from one another) cannot be reasonably treated as though these very parts are joined in a common memory by the time the execution of the loading (loading in 2 phases) takes place.

(ii) when the processor is construed for loading, the loading can be interpreted as transferring content from memory space in the card onto the target computer, because in view of the cooperative effect between the detection element and the basic module, i.e. the UI module can be construed as being loaded onto the computer by way of such cooperation. Reciting that the processor is 'configured' to load does not entail that the loading is already actualized; rather this indicates that such loading step is necessarily contingent to the triggering effect of a coupling event. And reciting that both parts of a same product (UI software) are stored on the memory of or in the electronic device can lend to many interpretations, thus, indefiniteness. Clearly, there is no relationship between how the parts are stored in relation to the processor being 'configured for' loading, i.e. storing indicating a **established** result of a transfer or loading whereas the very fact of loading has never been clear from the way the claim language is presented, wherein every single action is yet to be based on a coupling event.

One skill in the art would not being able to make use of the invention when there is no definite way to implement the above unclear limitation.

The broadest interpretation is that by way of the action taken by the basic module (first part of the UI software) responsive to the coupling event, the loading of the UI software module would enable the UI module (second part of the UI software) to be also stored in the environment of the electronic device

Claims 8-13, 16-20, 29, 22-27, 30, 31 are also rejected for failing to remedy to the above deficiency.

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5.   Claims 1-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shih et al., USPN: 6,405,362 (hereinafter Shih), in view of Garney, USPN: 5,319,751 (hereinafter Garney).

As per claim 1, Shih discloses a method for loading the software application from an expansion card in an electronic device (e.g. col. 3, lines 8-18), which expansion card can be coupled in a releasable manner to the device (Fig. 1; 3); wherein the method comprises:

starting the application software stored in the electronic device in at least two phases, and wherein the first phase includes starting the basic module (e.g. *event monitor* , step 300 - Fig. 2; col. 6, lines 25-37), and the second phase includes

detecting the coupling of the expansion card to the electronic device, and starting the application software module when the coupling of the expansion card to the electronic device is detected (e.g. Fig. 3), the detecting comprising:

transmitting a signal to the basic module indicating that coupling of the expansion card to

the electronic device is detected (Note: event monitor to see if insertion or removal event occurs

reads on module being already loaded, operating as to monitor signal coming from insertion

event, i.e. signal received by the monitoring process), and the basic module starting (e.g. Fig. 3)

the application software module when the signal is received.

Shih does not explicitly disclose wherein the application software comprises at least a

basic module and a user interface module, said basic module and said user interface module

being separate parts of the same application software prior to starting the application software.

However, Shih discloses software applications on hand held devices or user event-driven

applications, each of them necessarily include user interface modules (e.g. col. 6, lines 5-32);

hence has disclosed that the expansion software application to be loaded can be an user interface

software or application modules used in such handheld devices.

But Shih does not explicitly specify that said user interface software from the expansion

card is divided in a basic module and a user interface module, said basic module and said user

interface module being separate parts of the same user interface software. The loading and

activation of software from removable media being divided into a basic operating system level

module and a main software module was a known concept in the art of computer booting and

loading of operating system components or upgradable software. And this has been applied to

software provided via removable device being coupled to host computer system targeted for the

purpose to load up thereto such additional software or O.S. components. Whereas Shih uses a

shell stand-alone code or thread to snoop on card insertion event, the importance of straining on

the operating system resources or obviating tight dependency on architecture diversity and crash

due to conflict thereof is evidenced (see Shih: col. 6, line 41 to col. 7, line 18; col. 30-46; col. 9, line 24-53) via provision by the card insertion with module or OS related code to support the specifics of the software to install with such diversity and compatibility of resources. Analogous to the endeavor so concerned by Shih, Garney, in a method to activate/configure a processing device with software loaded from the removable resources being attached thereon analogous to the expansion card loading by Shih, teaches the software loading being done in 2 parts, with both parts being separate components of the software product; the first part being a stub for setting basic operating system configuration for preparing the host machine to incorporate the second part on the software which is loading and activation the content of the software in the removable card (Fig. 6-12). At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to provide a software product for such product loading method as suggested by Shih with a 2 separate parts or components just as taught by Garney just so to prepare the host target system with first software basic (Garney's stub) component to incorporate a second and main component of the software for which more resources are required and yet assure secure control and operation of the incorporation of this main software component in that less resources would be used to forestall conflicts by which more resources could be otherwise jeopardized.

Shih does not explicitly specify that said user interface software from the expansion card is divided in a basic module and a user interface module, said basic module and said user interface module being separate parts of the same user interface software. The loading and activation of software from removable media being divided into a basic operating system level module and a main software module was a known concept in the art of computer booting and

loading of operating system components or upgradable software. And this has been applied to

software provided via removable device being coupled to host computer system targeted for the

purpose to load up thereto such additional software or O.S. components. Whereas Shih uses a

shell stand-alone code or thread to snoop on card insertion event, the importance of straining on

the operating system resources or obviating tight dependency on architecture diversity and crash

due to conflict thereof is evidenced (see Shih: col. 6, line 41 to col. 7, line 18; col. 30-46; col. 9,

line 24-53) via provision by the card insertion with module or OS related code to support the

specifics of the software to install with such diversity and compatibility of resources. Analogous

to the endeavor so concerned by Shih, Garney, in a method to activate/configure a processing

device with software loaded from the removable resources being attached thereon analogous to

the expansion card loading by Shih, teaches the software loading being done in 2 parts, with both

parts being separate components of the software product; the first part being a stub for setting

basic operating system configuration for preparing the host machine to incorporate the second

part on the software which is loading and activation the content of the software in the removable

card (Fig. 6-12). At the time the invention was made, it would have been obvious to a person of

ordinary skill in the art to provide a software product for such product loading method as

suggested by Shih with a 2 separate parts or components just as taught by Garney just so to

prepare the host target system with first software basic (Garney's stub) component to

incorporate a second and main component of the software for which more resources are required

and yet assure secure control and operation of the incorporation of this main software component

in that less resources would be used to forestall conflicts by which more resources could be

otherwise jeopardized.

As per claim 2, Shih further teaches a method wherein the first basic module controls the execution of the second phase (step 315 – Fig. 3).

As per claim 3, Shih further teaches application programming interface and device driver to arrange communication between user interface software and expansion card (Fig. 3 – Note: O.S. device driver recognition cooperating with shell or windows API for signaling an hardware insertion is implicitly disclosed); said basic module being signaled on the coupling of the expansion card from device driver for effecting the loading and start-up of the software user interface module (e.g. col. 6, line 20 to col. 8, line 32).

As per claim 4, Shih does disclose wherein coupling an expansion card to a electronic device an interrupt signal is produced with OS examination of cause therefor; and information on the coupling is transmitted to a device driver (e.g. col 7, lines 30-50 – Note: Shell notifying a event monitor to allocate immediate resource for handling a signal is equivalent to interrupt signal handler for addressing hot insertion of card).

As per claim 5, Shih discloses wherein the decoupling of an expansion card halts processing of a user interface module without interrupting the basic module (Fig. 2,3 – Note: event monitor is required to remain functional even if card is removed for signal removal event, because Shih's monitor is not purported to be terminated as soon as one event gets detected).

As per claim 6, Shih discloses that memory is allocated for a user interface module when said module is loaded and said memory is deallocated when an expansion card removed from an electronic device (e.g. col. 7, lines 19-23, 62-67)

As per claim 7, Shih discloses a electronic device comprising

a program loader configured to load a expansion card application software in an electronic device (e.g. Fig. 1; col. 4, line 33 to col. 5, line 7; col. 6, line 5-55 – Note: computer system with monitoring event module reads on a monitoring process previously loaded and operating for detecting insertion of expandable device/insertable card);

a connecting element for coupling an expansion card in a releasable manner in electronic device; and

a processor to load, start and executing program modules in the device (e.g. Fig. 2,3); the processor configured to load the one application module of application software after the expansion card is coupled to the electronic device, the basic module already loaded (e.g. *event monitor* , step 300 - Fig. 2; col. 6, lines 25-37);

wherein a detecting element is configured to send a signal to the basic module when attachment of the expansion card to the electronic device is detected (Note: event monitor to see if insertion or removal event occurs reads on module being already loaded, operating as to monitor signal coming from insertion event, i.e. signal received by the monitoring process), wherein the basic module is configured to start the application software module (e.g. Fig. 3) when the signal is received.

From claim 1, Shih discloses that the application module of software application to be loaded can be a user interface software or application modules (e.g. col. 6, lines 5-32) used in such handheld devices.

However Shih does not explicitly disclose that the user interface software comprises a basic module and a user interface module; said basic module and said user interface module

being separate parts of the same user interface software and stored in a memory of the electronic

device; but this limitation has been addressed in claim 1 as being obvious in view of Garney.

As per claims 8-9, these are the apparatus claims corresponding to the method of claims

2-3, respectively. The claims are rejected under the same arguments as cited above, with Column

2, Line 1 referencing the apparatus (information process apparatus).

As per claims 10-11, these claims represent an apparatus performing a method

corresponding to the method of claims 3 and 4, hence are rejected using the same arguments as

cited above in the respective claims (Fig. 3 – Note: O.S. device driver recognition cooperating

with shell or windows API for signaling an hardware insertion is implicitly disclosed; col. 6, line

20 to col. 8, line 32; col 7, lines 30-50).

As per claim 12, Shih discloses communications with bus and multi-machine

environment (Fig. 1); and Garney teaches that the expansion card comprises a

transmitter/receiver unit and power amplifier (e.g. *device driver* - col.1, li.60 to col.2, li.4). At

the time of the invention, it was a well-known concept to one of ordinary skill in the art that a

power amplifier is commonly used in the output stage of a signal-producing device to isolate

output impedance. Additionally, it was also well-known in the art that a driver acts as

transmitter/receiver unit to control components of a specific computer resource, and that card

like modem or game adapter card come with a speaker being amplified by a power amplifier.

Hence if Garney (in combination with Shih) does not already provide a high frequency power

amplifier at the output stage of the transmitting unit, it would have been obvious to a person of

ordinary skill in the art to modify Garney's expansion card so that it does come with one such

amplifier to generate audio or signal with frequencies capable of being amplified for securing

distance transmission of signal or impedance matching purposes.

As per claim 13, Shih further teaches an apparatus for performing the method of claim 1

wherein the electronic device is a data processor (e.g. col. 6, lines 5-32).

As per claim 14, Shih teaches an apparatus comprising storage (e.g. *memory* - col.2, li.2;

Fig. 12) for storing user application software of a expansion card in an electronic device; a

processor configured to load, start and execute program modules (e.g. Fig. 1; col. 4, line 33 to

col. 5, line 7; col. 6, line 5-55), which expansion capable of being coupled to the electronic

device in a releasable manner;

wherein a loading program comprises procedures for loading the user application

software in at least two phases, wherein the first phase includes starting the basic module (e.g.

event monitor , step 300 - Fig. 2; col. 6, lines 25-37), and the second phase is executed when the

expansion card is coupled to the electronic device, the basic module already started (e.g. *event*

monitor , step 300 - Fig. 2; col. 6, lines 25-37 – Note: monitor a inserting event by a monitor

module reads on basic module already started);

wherein a detecting element is configured to send a signal to the basic module when

attachment of the expansion card to the electronic device is detected (Note: event monitor to see

if insertion or removal event occurs reads on module being already loaded, operating as to

monitor signal coming from insertion event, i.e. signal received by the monitoring process),

wherein the basic module is configured to start the application software module (e.g. Fig. 3)

when the signal is received by the basic module (Note: event monitor to see if insertion or

removal event occurs reads on insertion signal received by the monitoring process).

From claim 1, Shih discloses that the user application software to be loaded can be an

user interface software or user interface application modules (e.g. col. 6, lines 5-32) used in such

handheld devices.

However Shih does not explicitly disclose that the user interface software comprises at

least a basic module and a user interface module, said basic module and said user interface

module being separate parts of the same user interface software and stored in a memory of the

electronic device; but this limitation has been addressed in claim 1 as being obvious in view of

Garney.

As per claim 15, Shih discloses a method for loading the application software of an

expansion card in an electronic device, which expansion capable of being coupled to the

electronic device in a releasable manner(Fig. 1; 3); wherein the method comprises

executing the loading of the application software in at least two phases (Fig. 2, 3);

wherein the first phase includes loading and starting the basic module (e.g. *event monitor,*

step 300 - Fig. 2; col. 6, lines 25-37); and

the second phase includes detecting the coupling of the expansion card to the electronic

device and loading and starting the software application module (e.g. steps 315-320 – Fig. 3)

when the expansion card is coupled to the electronic device (Fig. 3), the detecting comprising

transmitting a signal to the basic module indicating that coupling of the expansion card to the

electronic device is detected (Note: event monitor to see if insertion or removal event occurs

reads on module being already loaded, operating as to monitor signal coming from insertion

event, i.e. signal received by the monitoring process or basic module).

From claim 1, Shih discloses that the expansion card software application to be loaded

can be a user interface software or application modules (e.g. col. 6, lines 5-32) used in such

handheld devices.

However Shih does not explicitly disclose that the user interface software comprises at

least a basic module and a user interface module, said basic module and said user interface

module being separate parts of the same user interface software and stored in a memory of the

electronic device; but this limitation has been addressed in claim 1 as being obvious in view of

Garney.

As per claims 16-20, refer to respective rejections of claims 2-6.

As per claim 21, this claim corresponds to claim 7, for it includes program loader,

connecting element, processor, and the user module and the basic module as recited therein;

hence is rejected using the corresponding rejections as set forth therein (except the detecting

element of claim 7).

As per claims 22-27, refer to respective rejections of claims 8-13.

As per claim 28, this claim corresponds to claim 14, for it includes storage, a processor,

a loading program, a user interface module and a basic as recited therein with the basic module

already loaded when the expansion card coupling is detected; hence is rejected using the

corresponding rejections as set forth therein (except the detecting element of claim 14).

As per claims 29 and 31, Shih does not explicitly disclose stopping the loading between

the first phase and the second phase, but this teaching is analogous to a user interface involving a

coupling of an expansion card, wherein the user can choose to interrupt the interface

communication for software loading anytime; that is, (see Fig. 3) any time the electronic device's

user chooses to stop the event monitor at stage 300, the user can effect an manual interruption

and enable the stop between the 2 phases, i.e. alternative by user to stop the installation before it

starts reads on optionally stopping. The limitation would have been intrinsic to any computer

user interaction or obvious because at the time the invention was made it was very well-known

concept that the computer user can anytime elect when to proceed with the any computer

process, e.g. loading process -- or to interrupt, hence making the feature as (being able) to

interrupt said recited loading steps an obvious and helpful feature; i.e. the user being given the

right or dynamic capability of his/her decision making at any moment of an installation via use

of an expansion card.

As per claim 30, the loading program being configured to stop the loading falls under the

intervention of a computer user who wishes to stop the loading; hence the stopping capability is

disclosed by virtue of claims 29-31.

Response to Arguments

6.   Applicants' arguments filed 10/11/2006 have been fully considered but they are not

persuasive. Following are the Examiner's observation in regard thereto.

(A)   Applicants have submitted that claim 1 does not recite loading software application from

an expansion card, as stated by the Examiner because Applicant recites that the user interface

software is stored in the electronic device (Appl. Rmrks, pg. 14, top). There is no specificity in

the claim about how the 'stored in' limitation is implemented particularly with respect to which

step of the installation, relative to the a timeframe in which processor starts executing or loading

software. There is no time-specific insight between the two limitations recited as 'starting ...

interface software ... in at least two phases' and 'wherein ... said basic module and said user

interface module being separate parts of the same ... prior to starting the user interface software'.

The mere fact that 2 modules are part of same software prior to the starting step does not enforce

any particular scenario that would be far distinguishing from the scenario wherein a basic

module is purported to communicate with a expansion card (before the expansion card insertion)

such that when a computer is dynamically coupled with the expansion card, the 2 modules are

still separate parts of the same software targeted for the loading; which is basically what Shih

discloses in light of Garney. That is, there is insufficient teaching concerning the time

relationship between the loading of the user interface module and the fact that both modules

(basic and UI modules) are stored in the computer. Besides, there is no evidence that claim 1

recites that the basic module and the UI module have been already stored on the electronic

device by the time the insertion of the expansion card is detected; nor is there any recital that not

only the UI software of the expansion card is not stored in the expansion card but also is it

necessarily or basically installed on the electronic device way before the expansion card is

coupled. Based on broad interpretation of the claimed 'user interface software of an expansion

card' and the detection of a expansion card to trigger a loading of said UI software, the steps

taught by Shih leading to both expansion card software and basic modules being stored on a

computer have anticipated the loading in 2 phases as recited in claim 1, with the obvious feature

provided by Garney. Applicants' arguments fail to comply with 37 CFR 1.111(b) because they

amount to a general allegation that the claims define a patentable invention without specifically

pointing out how the language of the claims patentably distinguishes them from the references.

The Applicants assert that software are integrated in the electronic device prior to the coupling in

the second phase and are sequentially loaded irrelevant to the expansion card content; but from

the language of the claim such scenario does not exist by way of broad interpretation or

insufficient time relationship between the loading phase and the state or whereabouts of the UI

software at the very moment where the coupling occurs (e.g. claim 1 lacks any slightest teaching

on the storing state of the expansion card UI software when the loading and coupling are started

or executed). The arguments are largely non-convincing.

(B)   Applicants have submitted that the basic module is not same as event monitor which is

recited as part of the UI software, not a shell process (Appl. Rmrks, pg. 14, bottom, pg. 15 top).

In response to Applicants'arguments against the references individually, one cannot show non-

obviousness by attacking references individually where the rejections are based on combinations

of references. See In re Keller, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); In re Merck & Co.,

800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). If the role of the basic module is such that the

basic module is part of the UI application (target software to load) so that it effectuate both the

card insertion detection and loading of said software onto the device, then Shih's process can be

enhanced to include the teachings by Garney whereby the expansion software in the card has a

first module being loaded to operate the steps recited in the second phase. The 103(a) rejection is

purported to address such rationale, given the motivation and reasons to combine. To rebut

Shih's, the Applicant is burdened to address the rejection as a whole not a piecemeal analysis of

reference taken separately.

(C)   Applicants have submitted that the basic and UI module are 'stored within the electronic

device', and that it is the expansion card that does not have this software (Appl. Rmrks, pg. 15,

middle); these amount to allegation without any explicit support of whatsoever from the

language in claim 1. The arguments that Shih's *autorun* being --in the removable medium-- not

what was claimed because recited is that both the basic module and the UI module are *stored*

within the electronic device (pg. 16, top) appear to be wrongly based on a limitation that is not

claimed. Not until claim 14 will the above limitation be claimed (see *).

(D)   Applicants have submitted Garney's stub is not the same as the claimed basic module

which is stored in the electronic device (Appl. Rmrks, pg. 16, bottom, pg. 17, top). Claim 1

does not provide a single and clear teaching that the basic module AND the UI module are

resident to the electronic device at the start of or at the very time the second phase executes the

UI software loading. Besides Garney is brought to fulfill the limitation that requires that

sometime 2 modules belonging to a same software product can operate in a different timeframe

yet fulfilling a same purpose to activate the intended software product to be coupled and loaded

into the target system. In response to Applicants' arguments against the references individually,

one cannot show non-obviousness by attacking references individually where the rejections are

based on combinations of references. See In re Keller, 642 F.2d 413, 208 USPQ 871 (CCPA

1981); In re Merck & Co., 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Applicants fail to

show why by providing 2 modules for a same product as approached by Garney would

jeopardize the software installation or loading by Shih's expansion card detection.

(E)   Applicants have submitted that Garney's full device driver remains resident on the card

and not transferred to the device (Appl. Rmrks, pg 17, middle). The argument has been

addressed at length in section B of the previous Office Action (mailed 7/06), and amounts to

mere assertion based on the teaching of one reference only, which is not proper in light of the

USC 103(a) type of rejection. In short, the Applicants' argument about the driver being executed

while resident on the card is therefore not persuasive.

(*) For the sake of argument, even if the limitation referred to as 'stored within the memory of the electronic device' is recited (see claims 7, 14-15, 21, 28), there is still no sufficient teaching from the claims that dictates any form of storage being accomplished as the start of any processor process or step of loading based on a detection element. The claim by reciting a processor 'configured to' load, start or execute does not make it plainly clear that any such loading is being actually effected while a (device) memory content has already stored within itself both the basic module and the UI software module. There is lack of relationship in time and space between the 'stored within ... electronic device' and any actual act of start to load or execute – a processor configured to do something only entails a possibility not a real action being taken. Thus, the 'stored within' limitation has been broadly interpreted as via the loading based on the detection both the basic and UI modules will end up as being stored within the memory of the device. Claim 1 never recites that 'stored within' limitation.

Besides, the stub by Garney is purported to address the limitation that the UI software has 2 parts, one of which being activated first to detect the insertion of the card and enable the loading of the UI module. Shih is not exactly providing a same software product with 2 parts cooperating to enable the loading of the main part after the first part has been activated. Garney provides one same software product having 2 parts (which Shih does not show), one of which being activated first to enable the loading of the second part phase. Besides, the claim does not make it clear that the loading of the basic module has to be done absolutely without the card being inserted at all. The newly added limitation recited as 'and being stored within the electronic device' is not giving any definite clue about where the basic module is located when the loading starts in the first phase -- when both modules, this module and the UI module, are

recited as being stored together (see USC 112, 2nd para); nor does the claimed 'loading and start-up of the basic module' in the 1st phase necessarily precludes any card insertion at the time of such loading(e.g. such that the card can be re-inserted later at a time where the basic module has already been running for the second phase to be enabled). The rejection has set forth the endeavor and benefits of such combination of analogous methodologies by both references. In regard to Applicants' arguments against the references individually, one cannot show non-obviousness by attacking references individually where the rejections are based on combinations of references.

(F)   Applicants have submitted that both Garney and Shih even as combined (Appl Rmrks, pg. 18, middle) do not provide all the teachings of the claimed invention because both the Applicants' basic and UI modules are loaded to the host device, not executed directly from the card. For one skill in the art, the terms such as 'upload' or 'download' in a software technologies are so common such that when a context involving a coupling of a expansion card, the loading of a module pertinent to such expansion card is construed as a loading from said card to the target device, the same manner as distribution software assets are downloaded or uploaded from or to other electronic devices. If loading means obtaining executable data from a slower storage medium to enable such data to be conveniently available, fetched thus expediently executed at a target runtime environment, the loading of software from a CD-ROM onto a memory of a target device reads on loading just like downloading data onto a Web browser level entails executable data being loaded for such Browser runtime. The claims lacking specifics or definiteness as to what exactly 'loading' really amounts to (Note: examples of indefiniteness would be the cases where the processor is only 'configured' to load, no loading data from any location to any

location is yet determined) in light of the UI software being pertinent to an expansion card in order to the claimed 'loading' to clearly distinguish from any loading of content from the card onto the target computer.

(G)   Applicants have submitted that Shih's event monitor (Appl. Rmrks pg. 19, middle) does not teach controlling execution of the UI module (in regard to claims 2-3). The rejection has set forth what controlling of the loading of the expansion card software module amounts to; and this has been met by Shih.

(H)   Applicants'argument about claim 5 (App. Rmrks, pg. 19) that Shih's event monitor is not same as the basic module of claim 1. The basic module is recited as detecting a coupling so to enable a loading as a second part of the 2 phases loading of an expansion card software, and there is nothing in this Applicants'feature that particularly distinguishes from Shih's module dedicated just to monitor such coupling, such that when a decoupling accidentally occurs, the module would still be in operating state. The argument is therefore not persuasive.

(I)   As for argument against claim 12 (App. Rmrks pg. 20, middle) in which it is argued that no reference teaches a transmitter/receiver as claimed. It is noted that there is no specifics in the claim to enforce a distinct implementation of transmitter or receiver that would clearly distinguish it from the modem or PCMCIA adapter as set forth in the rejection, notwithstanding the understanding that network communication frequencies are not low frequencies. Applicants'arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(J)   As per the added claims 29-31, Applicants have submitted that nowhere is there a Shih's

step in which a user can stop a loading between 2 phases as claimed (App. Rmrks, pg. 20,

bottom). The claim does not clearly establish what implementation is in place in order for not

allowing the user to intervene between the 2 phases of the process, rendering the argument (i.e.

user not coupling the card being the only way to stop) largely misplaced. The added claims

necessitated new grounds of rejection, and as set forth in the Office Action, the claims are

disclosed as intrinsic teachings or obvious, for the *stopping* imitation appears to be founded on a

well-known concept, and deemed intrinsic to Shih's computer usage, or obvious as set forth in

the Office Action.

Thus, the rejection will stand rejected as set forth.

Conclusion

7.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tuan A Vu
Patent Examiner,
Art Unit 2193
December 24, 2006